ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240567

Hard to Explain: On the Computational Hardness of In-Distribution Model Interpretation

Guy Amir[a,*,1], Shahaf Bassan[a,1] and Guy Katz[a]

[a]The Hebrew University of Jerusalem, Jerusalem, Israel

Abstract. The ability to interpret Machine Learning (ML) models is becoming increasingly essential. However, despite significant progress in the field, there remains a lack of rigorous characterization regarding the innate interpretability of different models. In an attempt to bridge this gap, recent work has demonstrated that it is possible to *formally* assess interpretability by studying the computational complexity of explaining the decisions of various models. In this setting, if explanations for a particular model can be obtained efficiently, the model is considered interpretable (since it can be explained "easily"). However, if generating explanations over an ML model is computationally intractable, it is considered uninterpretable. Prior research identified two key factors that influence the complexity of interpreting an ML model: (i) the *type* of the model (e.g., neural networks, decision trees, etc.); and (ii) the *form* of explanation (e.g., contrastive explanations, Shapley values, etc.). In this work, we claim that a third, important factor must also be considered for this analysis — the *underlying distribution* over which the explanation is obtained. Considering the underlying distribution is key in avoiding explanations that are *socially misaligned*, i.e., convey information that is biased and unhelpful to users. We demonstrate the significant influence of the underlying distribution on the resulting overall interpretation complexity, in two settings: (i) prediction models paired with an *external* out-of-distribution (OOD) detector; and (ii) prediction models designed to *inherently* generate socially aligned explanations. Our findings prove that the expressiveness of the distribution can significantly influence the overall complexity of interpretation, and identify essential prerequisites that a model must possess to generate socially aligned explanations. We regard this work as a step towards a rigorous characterization of the complexity of generating explanations for ML models, and towards gaining a mathematical understanding of their interpretability.

1 Introduction

Ensuring the interpretability of ML models is becoming increasingly vital, as it enhances their trustworthiness, particularly when deployed in safety-critical systems [23]. However, despite significant advancements in the field, there remains a notable lack of mathematical rigor in understanding the inherent interpretability of various ML models. For instance, many fundamental claims within interpretability, such as "decision trees are more interpretable than neural networks", are often regarded as folklore and lack sufficient mathematical rigor.

To bridge this gap, work by Barcelo et al. [6] proposes assessing the interpretability of an ML model by examining the computational complexity involved in generating various types of explanations for it. The idea is that if explanations can be efficiently obtained for an ML model, it can be considered interpretable. Conversely, if obtaining explanations is computationally intractable, the model is deemed uninterpretable. For example, while obtaining certain explanation forms for decision trees can be computed in polynomial or even linear time, these same tasks become NP-hard for neural networks [6, 26, 24]. This provides rigorous mathematical evidence that neural networks are indeed less interpretable than decision trees in these contexts.

The computational complexity of obtaining explanations was studied in a variety of different settings [6, 47, 9], in which the computational complexity is typically analyzed along two main axes: (i) the model *type* and (ii) the explanation *form*. For example, computing Shapley value explanations for decision trees can be obtained in polynomial time [3, 46], while obtaining minimum size contrastive explanations for neural networks is NP-complete [6].

The Distribution Component. In many explainability methods, understanding the rationale behind a specific input prediction often involves defining an explanation that satisfies certain properties in inputs similar to the one being interpreted. For instance, inputs that are identical to the original one in most features, with differences in only a few. This approach can be problematic because these new inputs might be out-of-distribution (OOD), and may deviate substantially from inputs of interest. Hence, the OOD inputs may affect the explanation in unexpected ways, and convey unintuitive information to users. Hase et al. [19] refer to explanations that disregard the input distribution as *socially misaligned*, i.e., convey information that is biased and unhelpful to users.

This general OOD phenomenon in explanations is termed "the OOD problem of explainability" [19] and is encountered in numerous explanation forms, including counterfactual explanations [37], contrastive explanations [50, 17], sufficient explanations [19, 50, 17], and Shapley values [42]. Therefore, many practical explanation techniques aim to mitigate the impact of OOD instances, making this a crucial aspect of computing more precise explanations [31, 11, 49, 19, 44, 53, 43].

In this work, we argue that evaluating the computational complexity of explaining the decision of a model, should not rely solely on the model type and the explanation form, but also on the *underlying distribution* over which the explanation is computed. The distribution component is crucial for ensuring that the computed explanations are socially aligned and meaningful. In this paper, we illustrate the impact of this factor on the overall interpretation complexity, in various

[*] Corresponding Author. Email: guy.amir2@mail.huji.ac.il.
[1] Equal contribution.

settings and scenarios.

A Running Example. Consider the task of classifying low-dimensional images as either “0" or “1". Due to the simplicity of this task, let us assume that it can be effectively learned using a simple decision tree classifier. Given an image classified as “0", we can interpret the prediction of the decision tree using a local, post-hoc explainability method. For instance, we can obtain a *sufficient reason* S [26, 13, 7]: a subset of features (in this case, pixels) that, when fixed, ensure the image remains classified as “0”, regardless of the assignment of the additional features $\overline{S}$. Fortunately, since this task was learned by a decision tree classifier, obtaining a locally minimal sufficient reason can be achieved in polynomial time [24].

However, despite their appeal, sufficient reasons, similarly to other explanation forms, suffer from the OOD problem of explainability [19, 50, 17]. In this particular case, the sufficient reason S may take into account *OOD assignments* over $\overline{S}$. In other words, setting the pixels of $\overline{S}$ to partial images that are OOD (e.g., images featuring unrelated digits, or cats) might result in the image being classified as “1”. This will preclude S from being a sufficient reason — even if it is one when taking into account only the context of interest (i.e., all in-distribution images of the digits “0” or “1”).

A common solution for bridging this gap is to train another model to detect OOD inputs, and then use it to dismantle the effect of any misleading assignment [31, 11, 49, 19, 44, 53]. However, the task of OOD detection is considered very challenging, both in theory [15, 36] and in practice [22, 40, 10] — as modeling the feature distribution is often harder than the original prediction task [42]. Hence, obtaining an OOD classifier may require training a very expressive model, such as a generative model that approximates the domain distribution $p_\phi(\mathbf{x})$. For our running example, for instance, learning to distinguish between in-distribution images (“0” or “1”) and OOD images (any other possible image) may be a substantially harder task than learning to classify images of “0” and “1”. Such a task may require the use of a much more expressive model, such as a deep generative neural network. The complexity of obtaining a sufficient reason S that ignores the effect of any OOD assignment may thus be much greater than that of simply explaining the decision tree classifier, without considering the distribution. Revisiting our running example, the findings in this study demonstrate that performing this task is indeed NP-hard, despite the fact that computing such an explanation without distribution alignment can be done in polynomial time.

Paper Structure. In Sec. 2, we start by covering the relevant background for this work. Next, in Sec. 3, we examine a wide variety of explanation forms, such as sufficiency-based, contrastive-based, and counting-based explanations, and study how they can be formalized to maintain social alignment. Specifically, we delve into the common scenario where the classification model is coupled with an additional component — an OOD detector. This detector plays a crucial role in mitigating the impact of OOD counterfactuals in explanations, and can be used to align various explanation forms with a distribution of interest. We proceed to demonstrate that diverse explanation forms can be unified through a single framework, which captures their shared structure. Given an OOD detector, this framework can be used to preserve the alignment of each of these explanations; as well as to study the computational complexity of obtaining them.

In Sec. 4 we prove that for any explanation matching our abstract form, the complexity of interpreting a model is dominated by the complexity of interpreting an OOD detector for the same type of explanation. Since OOD detection is computationally hard [15], the task of obtaining an *aligned* interpretation of the model may be substantially more complex than the misaligned form.

In Sec. 5 we study the specific case of *self-aligned* explanations. Here, our focus shifts from relying on an external OOD-detection model to the possibility of utilizing a single model that derives aligned explanations. Specifically, we focus on the case of efficiently producing a single model that serves both as a classifier and as an OOD detector, given that each of these is realized separately by the same model class. As we prove, this capability correlates to the degree of expressiveness inherent in various ML models — while some model types possess the required level of expressiveness, others do not. We prove these insights for specific model types and show that, assuming P$\neq$NP, both neural networks and decision trees have the capability to derive self-aligned explanations, while linear classifiers do not.

Furthermore, related work is covered in Sec. 6. We conclude in Sec. 7, and discuss the limitations of our theoretical framework, as well as potential future work in Sec. 8.

Due to space limitations, we provide only concise overviews of the proofs of our various claims, and refer the reader to the extended version of our paper [1] for the comprehensive and more detailed proofs.

2 Preliminaries

Domain

We assume a set of n features $\mathbf{x} = (x_1, \ldots, x_n)$, where the domain of each feature is $x_i \in \{0, 1\}$. The entire feature space is denoted as $\mathbb{F} = \{0, 1\}^n$. We seek to *locally* interpret the prediction of a binary classifier $f : \mathbb{F} \to \{0, 1\}$, i.e., given an input $\mathbf{x} \in \mathbb{F}$, to explain the prediction $f(\mathbf{x})$ of the classifier over this specific input. We follow common practice in the field, and concentrate on Boolean input and output values, to make the presentation clearer [2, 47, 6]. However, many of our findings are also applicable to scenarios involving real-valued data.

Complexity Classes and Second-Order Logic (SOL)

The paper assumes basic familiarity with the common complexity classes of polynomial time (PTIME) and nondeterministic polynomial time (NP, co-NP). The second order of polynomial hierarchy, i.e., Σ_2^P, which is briefly mentioned in the paper, is the set of problems that become members of NP given an oracle that solves co-NP problems in $O(1)$. We also discuss the class #P, which corresponds to the total number of accepting paths of a polynomial-time nondeterministic Turing machine. It is widely believed that PTIME$\subsetneq$ NP$\subsetneq$ $\Sigma_2^P \subsetneq$ #P [5]. We use the common convention $L_1 \leq_p L_2$ to denote a polynomial-time reduction from language L_1 to L_2, and $L_1 =_p L_2$ to indicate that such a reduction exists in both directions.

The paper also makes use of *second-order logic* (SOL) formulas — a generalization of the first-order predicate logic. In both logic forms, existential or universal quantifiers are applied to each variable or subset thereof, so that the formula evaluates to either true or false. However, we chose SOL formulas for our abstraction due to their high expressivity (in contrast to first-order logic queries suggested in [2]), as they can also encode an explanation size, which is infeasible with FOL. As such, SOL-based queries are rigorous enough to enable the formulation of general proofs that hold for any explanation within this framework. For each SOL formula Q, we define $\#Q$ as the corresponding counting problem over that formula — which counts the *number* of satisfying assignments for Q. Given a finite number of inputs, implying a finite logic-based model, each SOL formula is associated with a specific complexity class within the polynomial hierarchy, and with a corresponding counting class.

Explainability Queries

We follow prior work [6, 9] and define an *explainability query*, denoted Q, which represents some form of interpretation. Q takes both f and $\mathbf{x}$ as inputs, and it outputs information regarding the interpretation of $f(\mathbf{x})$. In line with previous work [34, 47, 2, 4, 9], our emphasis is on explainability queries that output an answer to a decision problem — providing a definite yes/no answer or, in the case of Q being a counting problem, a numerical value. For example, Q can provide a yes/no answer to the question *is a specific subset of features a sufficient reason?* It can also *count* the number of possible assignments in which the prediction is altered, or maintained.

Models

The techniques presented in this work are applicable to a diverse set of model classes. Still, we focus our attention on a few popular models, located at the extremities of the interpretability spectrum: decision trees, linear classifiers, and neural networks. Specifically, we address Free Binary Decision Diagrams (FBDDs), which serve as an extension of decision trees, along with Perceptrons and Multi-Layer Perceptrons (MLPs) employing ReLU activations. An exact formalization of these models appears in the extended paper [1].

3 Socially Aligned Explainability Queries

Context Indicator

To cope with the undesired effects of OOD input assignments, we consider some *context* $\mathbf{C} \subseteq \mathbb{F}$ over which an explanation is to be provided. Intuitively, context $\mathbf{C}$ denotes the entire potential set of in-distribution inputs that we take into consideration when providing an explanation, while disregarding the effect of any OOD assignment from $\mathbb{F} \setminus \mathbf{C}$. Because describing the context $\mathbf{C}$ explicitly is clearly non-trivial, in our framework, we instead assume the existence of a *context indicator* $\pi : \mathbb{F} \to \{0, 1\}$: a binary classifier over a specific context $\mathbf{C}$, i.e., $\pi(\mathbf{x}) = \mathbf{1}_{\{\mathbf{x} \in \mathbf{C}\}}$.

Naturally, assuming the existence of a context indicator π that perfectly captures the desired context $\mathbf{C}$ is non-trivial as well. For instance, in our running example, this requires π to identify any possible image of either "0" or "1". Nevertheless, practical tools were shown to be able to approximate such domains, for example, by using generative-model-based OOD classifiers, trained to learn the data distribution $p_\phi(\mathbf{x})$ [48, 32]. In these particular scenarios, the indicated $\mathbf{C}$ can be seen as a mere approximation of the true, intended context.

Socially Aligning Explainability Queries

Model interpretability is subjective, and this has led to the design of multiple forms of explanations in recent years. We focus here on a few widely used explanation forms, and analyze them rigorously.

Sufficiency-Based Explanations. A common definition of an explanation for a model f's decision with respect to an input $\mathbf{x}$ is that of a *sufficient reason* [26, 13, 7]. A sufficient reason is a subset of features $S \subseteq \{1, \ldots, n\}$ such that, when fixed to the corresponding values in $\mathbf{x}$, determine that the prediction remains $f(\mathbf{x})$, regardless of the other features' assignments [6, 34]. This notation is used quite often, and aligns with commonly used explainability techniques [38]. Formally put:

$$\forall(\mathbf{z} \in \mathbb{F}). \quad [f(\mathbf{x}_S; \mathbf{z}_{\bar{S}}) = f(\mathbf{x})] \tag{1}$$

where $(\mathbf{x}_S; \mathbf{z}_{\bar{S}})$ denotes an assignment in which the values of S are taken from $\mathbf{x}$ and, the remaining values (i.e., from $\overline{S}$), are taken from $\mathbf{z}$.

Given a context $\mathbf{C}$, indicated by π, a socially aligned sufficient reason is defined as follows [50, 17]:

$$\forall(\mathbf{z} \in \mathbb{F}). \quad [\pi(\mathbf{x}_S; \mathbf{z}_{\bar{S}}) = 1 \to f(\mathbf{x}_S; \mathbf{z}_{\bar{S}}) = f(\mathbf{x})] \tag{2}$$

A widely observed convention in the literature is that smaller sufficient reasons (relative to the size of $|S|$) are more meaningful than larger ones [26, 6, 18]. Consequently, it is interesting to consider *cardinally minimal sufficient reasons*. Clearly, these can also be obtained with respect to π. This leads us to our first explainability query:

MSR (Minimum Sufficient Reason):
Input: Model f, input $\mathbf{x}$, context indicator π, and integer k.
Output: *Yes*, if there exists a sufficient reason S for $f(\mathbf{x})$ with respect to π such that $|S| \leq k$, and *No* otherwise.

We note that we can consider the case of socially misaligned queries as a trivial case of this definition, in which the context indicator is the constant function $\pi := \mathbf{1}$, indicating the entire input space as in-distribution.

Contrastive/Counterfactual-Based Explanations. A different approach to interpreting a model is by observing subsets of features that, when altered, may cause the classification of the model to change [26, 6]. These are referred to as *contrastive explanations* or *contrastive reasons*, and the corresponding values are referred to as *counterfactual explanations*. We define a subset $S \subseteq \{1, \ldots, n\}$ as contrastive if altering its values may cause the original classification $f(\mathbf{x})$ to change:

$$\exists \mathbf{z} \in \mathbb{F}. \quad [f(\mathbf{x}_{\bar{S}}; \mathbf{z}_S) \neq f(\mathbf{x})] \tag{3}$$

To avoid counterfactual OOD assignments, a contrastive subset S can be obtained with respect to a context indicator π [50], by encoding:

$$\exists \mathbf{z} \in \mathbb{F}. \quad [\pi(\mathbf{x}_{\bar{S}}; \mathbf{z}_S) = 1 \land f(\mathbf{x}_{\bar{S}}; \mathbf{z}_S) \neq f(\mathbf{x})] \tag{4}$$

Similarly to sufficient reasons, smaller contrastive reasons tend to be more meaningful. Here, too, it is usually more informative to focus on cardinally minimal contrastive reasons, as expressed in the following explainability query:

MCR (Minimum Change Required):
Input: Model f, input $\mathbf{x}$, context indicator π, and integer k.
Output: *Yes*, if there exists some contrastive reason S such that $|S| \leq k$ for $f(\mathbf{x})$ with respect to π, and *No* otherwise.

Counting-Based Explanations. Finally, another common explainability form is based on exploring the number of assignment completions for maintaining (or altering) a specific classification [29, 14, 47]. As with previous explanation forms, we redefine the problem to avoid counting OOD completions, which may cause the social misalignment of the corresponding interpretation. In order to do so, we define the completion count c of S with respect to π as:

$$c(S) := |\{\mathbf{z} \in \{0,1\}^{|\overline{S}|}, \pi(\mathbf{x}_{\bar{S}}; \mathbf{z}_S) = 1, f(\mathbf{x}_{\bar{S}}; \mathbf{z}_S) \neq f(\mathbf{x})\}| \tag{5}$$

CC (Count Completions):
Input: Model f, input x, context indicator π, and subset of features S.
Output: The completion count $c(S)$ of $f(x)$ with respect to π.

The widely used Shapley values [42], which serve as a common form of explanation [33, 43], can also be characterized as a type of counting problem [46, 3].

Abstract Query Form

Many of the explanation forms studied in the literature, including the aforementioned ones, become more meaningful when the effect of OOD counterfactuals are reduced. For analyzing how distributions affect the complexity of obtaining explanations not only for one specific explanation, but for a wide array of explanation forms, we proceed to define *abstract* explainability queries. We then provide general results regarding the computational complexity of obtaining this abstract form of explanation.

The task of obtaining each of the explanation types discussed so far can be achieved by invoking a decision procedure for determining whether or not $f(\mathbf{x}_{\bar{S}};\mathbf{z}_S) = f(\mathbf{x})$ (or for solving the corresponding counting problem). These decision procedures receive a partial assignment $(\mathbf{x}_{\bar{S}};\mathbf{z}_S)$ of a given input $\mathbf{x}$, which fixes some features of $\mathbf{x}$ while allowing the rest to change according to an arbitrary $\mathbf{z}$; and their goal is to determine whether these assignments preserve, or alter, the classification outcome.

The task of deciding whether or not $f(\mathbf{x}_{\bar{S}};\mathbf{z}_S) = f(\mathbf{x})$ can, in turn, be formulated as an SOL formula, $SOL_{\neg f}$, which encodes that $f(\mathbf{x}_{\bar{S}};\mathbf{z}_S) \neq f(\mathbf{x})$ (or, again, the counting problem over that formula). If $SOL_{\neg f}$ is false, then the answer to the original problem is affirmative; and otherwise, it is negative.

The relevant formula is fully quantified, in a manner that represents a specific explanation form. For example, contrastive reason queries check whether there *exists* any assignment leading to a misclassification, whereas sufficient reason queries ask whether the classification stays constant for *all* possible completions. The goal is to eventually determine whether the formula is true or not, and equivalently — whether the explanation is correct.

In the extended paper [1], we show how each of the predefined explainability queries can be formalized using this notion, in which *MSR* (Minimum Sufficient Reason) and *MCR* (Minimum Change Required) are possible solutions to an underlying satisfiability query over $SOL_{\neg f}$, and *CC* (Count Completions) is the counting solution of $\#SOL_{\neg f}$. This can also be extended to additional explanation forms.

Definition 1. *Let $SOL_{\neg f}$ be an SOL formula encoding the query $f(\mathbf{x}_{\bar{S}};\mathbf{z}_S) \neq f(\mathbf{x})$. We define an* abstract query, $\mathbf{Q}$, *that receives f and $\mathbf{x}$ as inputs, and answers whether $SOL_{\neg f}$ is true. For the counting case, $\mathbf{Q}$ returns the counting of $\#SOL_{\neg f}$.*

Next, we adjust this abstract query form to provide only socially aligned explanations. This is performed by incorporating into the formula the additional constraint $\pi(\mathbf{x}_{\bar{S}};\mathbf{z}_S) = 1$, which guarantees that any explanation that satisfies the query is also in-distribution.

Definition 2. *Let $SOL_{\neg f,\pi}$ be an SOL formula encoding the query $f(\mathbf{x}_{\bar{S}};\mathbf{z}_S) \neq f(\mathbf{x}) \wedge \pi(\mathbf{x}_{\bar{S}};\mathbf{z}_S) = 1$. The respective aligned query, $\mathbf{Q}$, receives as inputs f, $\mathbf{x}$, and π, and answers whether $SOL_{\neg f,\pi}$ is true. For the counting case, $\mathbf{Q}$ returns the counting of $\#SOL_{\neg f,\pi}$.*

Any of the aligned query forms mentioned in the previous section can be described as an abstract notion of this query as we show in the extended paper [1]. In essence, this abstract form captures various (logically expressible) explanation formulations, over which we can dismantle the effect of OOD counterfactuals.

By using this single, broader form of $\mathbf{Q}$, we are able to prove general properties regarding socially aligned explainability queries, and deduce the complexity of interpreting these queries in various settings.

4 The Complexity of Obtaining Socially Aligned Explanations

A General Framework

To evaluate the computational complexity of interpreting a specific class of models, denoted as $\mathcal{C}_{\mathcal{M}}$, it is useful to define $Q(\mathcal{C}_{\mathcal{M}})$ as the computational problem represented by interpreting a set of models within the class $\mathcal{C}_{\mathcal{M}}$ with respect to an explainability query Q [6, 9]. To illustrate, let us consider the class of multi-layer perceptrons denoted as $\mathcal{C}_{\text{MLP}}$. MSR$(\mathcal{C}_{\text{MLP}})$ is then the computational problem of obtaining cardinally minimal sufficient reasons for an MLP, given an input $\mathbf{x}$.

While this formalization is helpful for assessing the interpretability of a specific model type, it does not consider the underlying context and thus, it may produce socially misaligned explanations.

We revisit our running example, where our model f represents a decision tree. We further assume that the decision of whether $\mathbf{x} \in \mathbf{C}$ (or equivalently, whether $\mathbf{x}$ is in-distribution) is learned by another model, e.g., a deep neural network. In this scenario, the context indicator π belongs to a different class than f (which is in $\mathcal{C}_{\mathcal{M}}$). In such a case, we should pose a different type of question that assesses the computational complexity of providing a socially aligned explanation for an instance classified by f. Specifically, we need to determine the computational complexity of interpreting a model $f \in C_M$ while ensuring its alignment with a context indicator function $\pi \in \mathcal{C}_\pi$. As mentioned earlier, in many instances (including our example), π corresponds to a more expressive function than f, potentially dominating the overall complexity. Therefore, we introduce the following notion that enables us to assess the computational complexity of models in $\mathcal{C}_{\mathcal{M}}$ with respect to a class of context indicators $\mathcal{C}_\pi$.

Definition 3. *Given an explainability query Q, a class of prediction models $\mathcal{C}_{\mathcal{M}}$, and a class of context indicators $\mathcal{C}_\pi$, we define $Q(\mathcal{C}_{\mathcal{M}}, \mathcal{C}_\pi)$ as the computational problem of Q defined by the set of functions within $\mathcal{C}_{\mathcal{M}}$, with respect to the contexts induced by the functions of $\mathcal{C}_\pi$.*

For our running example, $Q(\mathcal{C}_{\text{DT}}, \mathcal{C}_{\text{MLP}})$ denotes the computational complexity of some explainability query Q, given that our classification model is a decision tree and the OOD detector is a multi-layer perceptron. We note that, similarly to the previously studied evaluation of $Q(\mathcal{C}_{\mathcal{M}})$ [6], the formalization of $Q(\mathcal{C}_{\mathcal{M}}, \mathcal{C}_\pi)$ considers a "worst-case" scenario of the corresponding alignment, and not any parameter-specific configuration. This is captured by assessing the corresponding complexity with respect to a class of prediction models and a class of distribution indicators.

The Complexity of $Q(\mathcal{C}_{\mathcal{M}}, \mathcal{C}_\pi)$

We prove a connection between the complexity of calculating an aligned explanation $Q(\mathcal{C}_{\mathcal{M}}, \mathcal{C}_\pi)$, to the complexity of obtaining misaligned explanations of either $Q(\mathcal{C}_{\mathcal{M}})$ or $Q(\mathcal{C}_\pi)$. This relation holds in a broad sense, as we prove it for our abstract query form $\mathbf{Q}$, defined in Sec. 3.

First, clearly, if $\mathbf{1} \in \mathcal{C}_\pi$ ($\mathbf{1}$ is a trivial function that accepts any possible input as in-context), then $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$ is polynomially reducible from $\mathbf{Q}(\mathcal{C}_\mathcal{M})$. We note that $\mathbf{1} \in \mathcal{C}_\pi$ is a trivial request for any expressive class of context indicators, for example, assuming the existence of a neural network that always outputs 1.

Theorem 1. *If $\mathbf{1} \in \mathcal{C}_\pi$ then $\mathbf{Q}(\mathcal{C}_\mathcal{M}) \leq_P \mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$.*

This result is, of course, not surprising and a more interesting connection to explore is the less straightforward relation between $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$ and $\mathbf{Q}(\mathcal{C}_\pi)$. We show that a similar result to the former can be obtained in this case as well, provided that $\mathcal{C}_\pi$ is *symmetrically constructible* (given some $f \in \mathcal{C}_\pi$, we can construct in polynomial time $\neg f \in \mathcal{C}_\pi$); and that $\mathcal{C}_M$ is *naively constructible* (given some $\mathbf{x} \in \mathbb{F}$, it holds that we can construct in polynomial time $\mathbf{1}_{\{\mathbf{x}\}} \in \mathcal{C}_\mathcal{M}$). A full formalization of these conditions is provided in our extended paper [1]. Later in this section, we also demonstrate that these constructions also hold for popular function classes, and provide model-specific instantiations of our framework.

Theorem 2. *If $\mathcal{C}_\mathcal{M}$ is symmetrically constructible and $\mathcal{C}_\pi$ is naively constructible, then $\mathbf{Q}(\mathcal{C}_\pi) \leq_P \mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$.*

Theorem 2 indicates that, given basic assumptions regarding the expressivity of $\mathcal{C}_\mathcal{M}$ and $\mathcal{C}_\pi$, it holds that the complexity of evaluating $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$, i.e., interpreting a model from $\mathcal{C}_\mathcal{M}$ with respect to a model from $\mathcal{C}_\pi$, for some explainability query $\mathbf{Q}$, is *at least as hard* as interpreting $\mathbf{Q}(\mathcal{C}_\pi)$, i.e., interpreting the OOD detector π. This is significant — as in many cases $\mathcal{C}_\pi$, the class associated with the input distribution, is much more expressive than C_M, the class associated with the prediction model, and hence may be much harder to interpret.

Proof sketch. The reduction exploits the naive constructibility of $\mathcal{C}_\mathcal{M}$, with the aim of rendering obsolete the conjunct responsible for validating whether a subset is contrastive. The reduction takes advantage of the fact that $\pi \in \mathcal{C}_\pi$ is symmetrically constructible in order to transform π to validate the *model* instead of the indicated context. By employing this approach, it becomes feasible to polynomially reduce any SOL formula representing $\mathbf{Q}(\mathcal{C}_\pi)$ to an equivalent SOL formula under the formulation of $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$. Consequently, any decision or counting solution for the original SOL formula will be tantamount to solving an equivalent SOL formula corresponding to a query seeking socially aligned explanations.

Model-Specific Framework Instantiations

Next, we present specific results when focusing on FBDDs, Perceptrons, and MLPs. It is straightforward to show that these classes of models match our theoretical framework, as the following holds (and proven in our extended paper [1]):

Proposition 1. *FBDDs, Perceptrons, and MLPs are all symmetrically constructible and naively constructible.*

Dominance of Interpreting MLPs. We prove that when dealing with complexity classes of explainability queries that are from the polynomial hierarchy (such as NP, Σ_2^P, etc.), the complexity class associated with the MLP always dominates the overall complexity. Hence, the *exact* complexity class of $Q(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$ when $\mathcal{C}_\mathcal{M} = \mathcal{C}_{\text{MLP}}$ and/or $\mathcal{C}_\pi = \mathcal{C}_{\text{MLP}}$, is equivalent to that of $Q(\mathcal{C}_{\text{MLP}})$. This claim holds for any class of polynomially computable functions.

Theorem 3. *Let $\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi$ be classes of polynomially computable functions such that $\mathcal{C}_\mathcal{M} = \mathcal{C}_{MLP}$ or $\mathcal{C}_\pi = \mathcal{C}_{MLP}$. If $\mathbf{Q}(\mathcal{C}_{MLP})$ is $\mathcal{K}$-complete, where $\mathcal{K}$ is a complexity class of the polynomial hierarchy (or the class associated with its counting problem), then $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$ is also $\mathcal{K}$-complete.*

The "hardness" part of Theorem 3 is a direct consequence of Theorems 1 and 2. However, when specifically considering MLPs, completeness also holds. The proof of this claim is relegated to the extended paper [1], and is a result of the fact that any Boolean circuit can be polynomially reduced to an MLP [6]. This relation implies that the "hardest" possible complexity class in the polynomial hierarchy is always associated with the one for interpreting an MLP over $\mathbf{Q}$. Fig. 1 depicts the relations among different complexity classes, as derived from Theorems 1, 2, and 3.

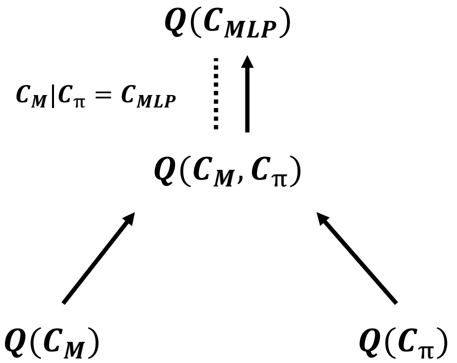

Figure 1: A visual illustration of Theorems 1, 2, and 3. Dashed lines depict that both queries are in the same complexity class, and are hard for that class. Arrows are directed from the query with the "easier" complexity class to the query with the "harder" complexity class.

In Table 1, we exemplify the aforementioned explainability queries (*MCR*, *MSR*, and *CC*) and a specific scenario where $\mathcal{C}_\mathcal{M}$ is set to either $\mathcal{C}_{\text{FBDD}}$ or $\mathcal{C}_{\text{Perceptron}}$, whereas the context indicator $\mathcal{C}_\pi$ is set to $\mathcal{C}_{\text{MLP}}$ (this is the case of our running example, in which the OOD detection is performed using a more expressive model than the original classifier). Hence, Theorem 3 implies that the complexity of solving the aligned query is primarily determined by the complexity involved in using an MLP, as summarized in Table 1.

5 "Self-Alignment": Incorporating Social Alignment within a Single Model

Until now, we focused on the general scenario in which f and π are chosen from two *different* model classes (for instance f is a decision tree, and π is a neural network). However, in some cases, f and π can be two models of the same type, i.e., from the same class. In this scenario, given a classifier and an OOD detector, both from the same class, practitioners might decide to train a single model that learns *both* the prediction task and the alignment task. More formally, we say that a single model class $\mathcal{C}$ is "self-aligned" when it is expressive enough to incorporate this dual procedure. This is demonstrated by the fact that given a model f and a context indicator π, a new model g can be efficiently constructed to show the alignment of f with respect to the distribution indicated by π:

Definition 4. *A class of models $\mathcal{C}$ is* self-aligned *if for any $f, \pi \in \mathcal{C}$, and any inputs $\boldsymbol{x}$ and I, there exists a polynomially constructible function $g \in \mathcal{C}$, such that:*

$$\langle f, \pi, \boldsymbol{x}, I\rangle \in \mathbf{Q}(\mathcal{C}, \mathcal{C}) \iff \langle g, \boldsymbol{x}, I\rangle \in \mathbf{Q}(\mathcal{C}) \quad (6)$$

Table 1: The computational complexity of $\mathbf{Q}(\mathcal{C}_\mathcal{M})$ and $\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_\pi)$ with respect to various explainability queries.

	$\mathcal{C}_\mathcal{M} = \mathcal{C}_{\textbf{FBDD}}$		$\mathcal{C}_\mathcal{M} = \mathcal{C}_{\textbf{Perceptron}}$	
	$\mathbf{Q}(C_M)$	$\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_{\textbf{MLP}})$	$\mathbf{Q}(\mathcal{C}_\mathcal{M})$	$\mathbf{Q}(\mathcal{C}_\mathcal{M}, \mathcal{C}_{\textbf{MLP}})$
MCR	PTIME	NP-complete	PTIME	NP-complete
MSR	NP-complete	Σ_2^P-complete	PTIME	Σ_2^P-complete
CC	PTIME	#P-complete	#P-complete	#P-complete

Intuitively, for any possible explainability query within $\mathbf{Q}$ (decision or counting), explanations of f, aligned by π, can be expressed by a single aggregated function g. Clearly, g must be at least as expressive as the original models f and π. This raises the question of how expressive a class of models $\mathcal{C}$ should be, for it to be self-aligned.

Theorem 4. *Given a class of models $\mathcal{C}$, if for any $f_1, f_2 \in \mathcal{C}$, we can polynomially construct $g := f_1[op]f_2 \in \mathcal{C}$, for $[op] \in \{\wedge, \rightarrow\}$, then $\mathcal{C}$ is self-aligned.*

Intuitively, classes of models that are capable of expressing the logical operators $\rightarrow$ and $\wedge$ are capable of "capturing" that a given explanation form is determined by its underlying distribution. The proof of this theorem is relegated to the extended paper [1], and can be obtained by showing an equivalence between the two underlying formalizations.

If self-alignment implies that, given a prediction model f and a context indicator π, we can attain a single aggregated model g — then clearly the computational complexity of interpreting $f \in C$ with respect to $\pi \in \mathcal{C}$ (i.e., the complexity of $\mathbf{Q}(\mathcal{C}, \mathcal{C})$) is correlated to the complexity of interpreting $g \in \mathcal{C}$ (i.e., the complexity of $\mathbf{Q}(\mathcal{C})$). This can be demonstrated by the subsequent proposition:

Proposition 2. *If the conditions in Theorem 4 hold for a class of models $\mathcal{C}$, then $\mathbf{Q}(\mathcal{C}, \mathcal{C}) =_P \mathbf{Q}(\mathcal{C})$.*

Model-Specific Results

We move on to analyze which of the aforementioned model classes incorporate self-alignment. First, we show that both FBDDs and MLPs are self-aligned, which is a result of their capability to polynomially express $\rightarrow$ and $\wedge$ relations within their class:

Proposition 3. *FBDDs and MLPs are self-aligned, and hence, it follows that: $\mathbf{Q}(\mathcal{C}_{FBDD}, \mathcal{C}_{FBDD}) =_P \mathbf{Q}(\mathcal{C}_{FBDD})$ and $\mathbf{Q}(\mathcal{C}_{MLP}, \mathcal{C}_{MLP}) =_P \mathbf{Q}(\mathcal{C}_{MLP})$.*

However, in contrast to decision trees and neural networks, linear classifiers lack the ability to capture the notion of self-alignment. It is important to note that a single Perceptron cannot inherently represent the $\rightarrow$ and $\wedge$ relations over two other Perceptrons. That said, it is worth emphasizing that this observation alone does not conclusively establish their lack of self-alignment, as this condition is sufficient but not necessary. To rigorously prove the inability of Perceptrons to be self-aligned, we prove the subsequent proposition:

Proposition 4. *While the query $MCR(\mathcal{C}_{Perceptron})$ can be solved in polynomial time, the query $MCR(\mathcal{C}_{Perceptron}, \mathcal{C}_{Perceptron})$ is NP-complete.*

Proof sketch. Membership results from the fact that we can guess a subset of features S and validate whether it is contrastive for f and whether it is also in-distribution (by feeding it to π). For hardness, we reduce from *SSP* (the k-subset-sum problem), which is a classic NP-complete problem. The reduction exploits the ranges of the Perceptrons of both f and π in order to bind the target sum T of the subset, both from above and from below.

Building upon Proposition 4, we can deduce the following corollary (proved in the extended paper [1]):

Theorem 5. *Assuming that $P \not\leq NP$, the class $\mathcal{C}_{Perceptron}$ is not self-aligned.*

These findings underscore a crucial aspect concerning the interpretability of Perceptrons. While producing explanations pertaining to them can be achieved with low computational complexity (providing further evidence of their interpretability), they are not self-aligned. Consequently, obtaining *aligned* explanations using Perceptrons necessitates the adoption of a more sophisticated model, that is expressive enough to incorporate social alignment — and this, in turn, can significantly increase the overall complexity of their interpretation.

6 Related Work

This work continues a line of research that focuses on *Formal XAI* [25, 47, 4, 2, 27, 7, 8]. Prior studies have already investigated the explanation forms that were analyzed within our work [2, 47, 4, 2], including sufficiency-based explainability queries (*MSR*) [26, 34], contrastive/counterfactual-based queries (*MCR*) [41, 28], and counting-based queries (*CC*) [14]. Other work [17] defined formal notions of sufficient and contrastive reasons under specific contexts and suggested ways to compute them on a wide range of models [50]. However, these explanation forms were not analyzed with respect to their overarching computational complexity. Closer to ours is the work of Cooper et al. [12] which analyzes different properties (including the computational complexity) of sufficiency-based explanations under logical constraints. We also acknowledge the work of Arenas et al. [2], which describes a general logic-based explanation form, similar to our abstract query form. While their work focuses on explanations of first-order logic forms for decision queries, our approach is more expressive, encompassing second-order logic forms that incorporate both decision-based and counting-based explanations.

Another line of research examines the computational complexity of obtaining Shapley value-based explanations [3, 46, 35], where alignment with respect to a given distribution is vital [42]. Specifically, Van den Broeck et al. [46] identify a complexity gap in interpreting Shapley values when considering fully factorized or *Naive Bayes-modeled distributions*.

In some cases, the term "sufficient reason" is also defined as an *abductive explanation* [26] and correlates with the notion of a *prime implicant* for a Boolean classifier [14]. The *CC* query is associated with probabilistic notions of explainability, by correlating the precision of the explanation with the number of possible input completions [38, 47]. A similar notion, formally known as a δ-*relevant set* [29, 47], focuses on bounding this specific portion.

The dependency of explanations on OOD assignments has been studied extensively [51, 43, 16, 20, 31, 21, 49, 42]. Specifically, many heuristic-based tools and frameworks have been proposed for dealing with the OOD counterfactual problem in model explainability. These include marginalizing the prediction of the model over possible counterfactual assignments [53, 31, 49], sampling points in the proximity of the original input [11, 39, 38], as well as *counterfactual training* [19, 45] — a method that, similarly to adversarial training [52], seeks to robustify models to OOD counterfactuals. Other work focuses on mitigating the effect of OOD assignments on the computation of Shapley values [42, 30, 44]. In spite of these notable accomplishments, the theoretical analysis of the OOD counterfactual problem with respect to its computational complexity has yet to be thoroughly examined.

7 Conclusion

Computational complexity theory stands as a potential avenue to formally assess the interpretability of various ML models. Prior research examined this by considering two main factors: the model type and the explanation form. We claim that a third and important factor should be taken into consideration — the underlying distribution over which the explanation is computed. To achieve this goal, we generalize existing explainability queries and show how a unified form can describe the desired social alignment requirement for any explanation form under our second-order logic formalization. Moreover, we present a framework for assessing the computational complexity of these queries and demonstrate that, for a broad range of model types and query forms, providing socially aligned explanations is as hard as interpreting a model designed to detect OOD inputs. As OOD detection is known to be substantially difficult, such models may often require more expressive capacity than the original classification models, significantly impacting the overall complexity of model interpretation. Finally, we provide an analysis of the required capacity of models to inherently produce aligned explanations without using an external OOD detector. We hope that our work serves as a foundation for a deeper mathematical understanding of the interpretability pertaining to various ML models.

8 Limitations and Future Work

Our framework can be extended along several different axes. First and foremost, we note that assuming the existence of a context indicator π for identifying OOD inputs is highly non-trivial. Previous work, both theoretical and practical, has highlighted the challenges associated with obtaining such an OOD detector [15, 36, 22, 40, 10]. However, it is important to emphasize that our framework does not necessarily assume the complete accuracy or correctness of such a classifier. Instead, π can be viewed as a function that provides an *approximation* of the underlying context $\mathbf{C}$. Therefore, future research endeavors could center around evaluating the computational complexity of specific approximations tailored to particular contexts of interest. While these approximations may only offer a *partially* guaranteed solution to the alignment issue, they may still exhibit an improved complexity overall.

Other limitations correspond to similar (non-aligned) approaches for analyzing the computational complexity of obtaining explanations [6, 47, 9]. Firstly, our analysis considers only a worst-case scenario that may change under various parameter-specific configurations. Secondly, the natural subjectivity of interpretability makes it challenging to analyze the computational complexity of interpreting a model in a single "correct" way. To address this issue, theoretical frameworks define various explainability queries and evaluate them separately. We regard our proof for a wide range of explainability queries $\mathbf{Q}$ (the abstract query form) as potential evidence that the shared characteristics among different types of explainability queries can be utilized to offer more generalized assessments.

Finally, we highlight that our study primarily concentrates on an OOD detector $\pi(\mathbf{x})$, which classifies each input as either in-distribution or OOD, rather than on the input distribution $p_\theta(\mathbf{x})$ itself. This approach is due to the strictly formal nature of the explanations we investigate; an explanation is either valid or not, necessitating a definitive categorization of the presence or absence of each input. In contrast, probabilistic explanation forms, such as δ-relevant sets [47, 29] or Shapley values [33, 42], are defined in relation to the distribution itself and can also be assessed based on the computational complexity of obtaining them. For instance, a recent study by Marzouk et al. [35] explores the computational complexity of calculating Shapley values within Markovian distributions. Future research can focus on expanding the strictly formal explanation framework discussed here to include probabilistic explanation forms as well, where complexity assessments would focus directly on the input distribution $p_\theta(\mathbf{x})$ rather than on the OOD detector $\pi(\mathbf{x})$. Other, broader future work can explore the relation between the computational complexity of generating explanations (our current focus) and the complexity of the explanations themselves. This can be achieved using various tools, such as Kolmogorov complexity. We also cover additional extensions of our framework in our extended paper [1].

Acknowledgments

This work was partially funded by the European Union (ERC, VeriDeL, 101112713). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The work of Amir was further supported by a scholarship from the Clore Israel Foundation.

References

[1] G. Amir, S. Bassan, and G. Katz. Hard to Explain: On the Computational Hardness of In-Distribution Model Interpretation, 2024. Technical Report. https://arxiv.org/abs/2408.03915.

[2] M. Arenas, D. Baez, P. Barceló, J. Pérez, and B. Subercaseaux. Foundations of Symbolic Languages for Model Interpretability. In *Proc. 34th Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, pages 11690–11701, 2021.

[3] M. Arenas, P. Barceló, L. Bertossi, and M. Monet. The Tractability of SHAP-Score-Based Explanations for Classification over Deterministic and Decomposable Boolean Circuits. In *Proc. 35th AAAI Conf. on Artificial Intelligence*, pages 6670–6678, 2021.

[4] M. Arenas, P. Barceló, M. Romero Orth, and B. Subercaseaux. On Computing Probabilistic Explanations for Decision Trees. In *Proc. 35th Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, pages 28695–28707, 2022.

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[6] P. Barceló, M. Monet, J. Pérez, and B. Subercaseaux. Model Interpretability through the Lens of Computational Complexity. In *Proc. 33rd Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, pages 15487–15498, 2020.

[7] S. Bassan and G. Katz. Towards Formal Approximated Minimal Explanations of Neural Networks. In *Proc. 29th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 187–207, 2023.

[8] S. Bassan, G. Amir, D. Corsi, I. Refaeli, and G. Katz. Formally Explaining Neural Networks within Reactive Systems. In *Proc. 23rd Int. Conf. on Formal Methods in Computer-Aided Design (FMCAD)*, pages 10–22, 2023.

[9] S. Bassan, G. Amir, and G. Katz. Local vs. Global Interpretability: A Computational Complexity Perspective. In *Proc. 41st Int. Conf. on Machine Learning (ICML)*, 2024.

[10] D. Berend, X. Xie, L. Ma, L. Zhou, Y. Liu, C. Xu, and J. Zhao. Cats are not Fish: Deep Learning Testing Calls for Out-of-Distribution Awareness. In *Proc. 35th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*, pages 1041–1052, 2020.

[11] C. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining Image Classifiers by Counterfactual Generation. In *Proc. 7th Int. Conf. on Learning Representations (ICLR)*, 2019.

[12] M. Cooper and L. Amgoud. Abductive Explanations of Classifiers under Constraints: Complexity and Properties. In *26th European Conf. on Artificial Intelligence (ECAI)*, 2023.

[13] A. Darwiche and A. Hirth. On the Reasons Behind Decisions. In *Proc. 23rd European Conf. on Artificial Intelligence (ECAI)*, pages 712–720, 2020.

[14] A. Darwiche and P. Marquis. A Knowledge Compilation Map. *Journal of Artificial Intelligence Research (JAIR)*, 17:229–264, 2002.

[15] Z. Fang, Y. Li, J. Lu, J. Dong, B. Han, and F. Liu. Is Out-of-Distribution Detection Learnable? In *Proc. 36th Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[16] R. Fong and A. Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3429–3437, 2017.

[17] N. Gorji and S. Rubin. Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints. In *Proc. 36th AAAI Conf. on Artificial Intelligence*, pages 5660–5667, 2022.

[18] J. Halpern and J. Pearl. Causes and Explanations: A Structural-Model Approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 2005.

[19] P. Hase, H. Xie, and M. Bansal. The Out-of-Distribution Problem in Explainability and Search Methods for Feature Importance Explanations. In *Proc. 34th Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, pages 3650–3666, 2021.

[20] S. Hooker, D. Erhan, P. Kindermans, and B. Kim. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Proc. 32nd Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[21] C. Hsieh, C. Yeh, X. Liu, P. Ravikumar, S. Kim, S. Kumar, and C. Hsieh. Evaluations and Methods for Explanation through Robustness Analysis. In *Proc. 9th Int. Conf. on Learning Representations (ICLR)*, 2021.

[22] Y. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi. A Survey of Safety and Trustworthiness of Deep Neural Networks: Verification, Testing, Adversarial attack and Defence, and Interpretability. *Computer Science Review*, 37:100270, 2020.

[24] X. Huang, Y. Izza, A. Ignatiev, and J. Marques-Silva. On Efficiently Explaining Graph-Based Classifiers, 2021. Technical Report. https://arxiv.org/abs/2106.01350.

[25] A. Ignatiev. Towards Trustable Explainable AI. In *Proc. 29th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 5154–5158, 2020.

[26] A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-Based Explanations for Machine Learning Models. In *Proc. 33rd AAAI Conf. on Artificial Intelligence*, pages 1511–1519, 2019.

[27] A. Ignatiev, N. Narodytska, and J. Marques-Silva. On Relating Explanations and Adversarial Examples. In *Proc. 32nd Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[28] A. Ignatiev, N. Narodytska, N. Asher, and J. Marques-Silva. From Contrastive to Abductive Explanations and Back Again. In *Proc. Int. Conf. Italian Association for Artificial Intelligence*, 2020.

[29] Y. Izza, A. Ignatiev, N. Narodytska, M. Cooper, and J. Marques-Silva. Efficient Explanations with Relevant Sets, 2021. Technical Report. https://arxiv.org/abs/2106.00546.

[30] D. Janzing, L. Minorics, and P. Blöbaum. Feature Relevance Quantification in Explainable AI: A Causal Problem. In *Proc. 23rd Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pages 2907–2916, 2020.

[31] S. Kim, J. Yi, E. Kim, and S. Yoon. Interpretation of NLP Models Through Input Marginalization. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[32] C. Liang, P. Huang, W. Lai, and Z. Ruan. GAN-Based Out-of-Domain Detection Using Both In-Domain and Out-of-Domain Samples. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7663–7667, 2021.

[33] S. Lundberg and S. Lee. A Unified Approach to Interpreting Model Predictions. In *Proc. 30th Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[34] J. Marques-Silva, T. Gerspacher, M. Cooper, A. Ignatiev, and N. Narodytska. Explaining Naive Bayes and Other Linear Classifiers with Polynomial Time and Delay. In *Proc. 33rd Int. Conf. on Advances in Neural Information Processing Systems (NeurIPS)*, pages 20590–20600, 2020.

[35] R. Marzouk and C. de La Higuera. On the Tractability of SHAP Explanations under Markovian Distributions, 2024. Technical Report. http://arxiv.org/abs/2405.02936.

[36] P. Morteza and Y. Li. Provable Guarantees for Understanding Out-of-Distribution Detection. In *Proc. 36th AAAI Conf. on Artificial Intelligence*, pages 7831–7840, 2022.

[37] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach. FACE: Feasible and Actionable Counterfactual Explanations. In *Proc. AAAI/ACM Conf. on AI, Ethics, and Society (AIES)*, 2020.

[38] M. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-Precision Model-Agnostic Explanations. In *Proc. 32nd AAAI Conf. on Artificial Intelligence*, 2018.

[39] S. Sanyal and X. Ren. Discretized Integrated Gradients for Explaining Language Models. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[40] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. Núñez, and J. Luque. Input Complexity and Out-of-Distribution Detection with Likelihood-Based Generative Models. In *Proc. 7th Int. Conf. on Learning Representations (ICLR)*, 2019.

[41] A. Shih, A. Choi, and A. Darwiche. Formal Verification of Bayesian Network Classifiers. In *Proc. Int. Conf. on Probabilistic Graphical Models (PGM)*, pages 427–438, 2018.

[42] M. Sundararajan and A. Najmi. The Many Shapley Values for Model Explanation. In *Proc. 37th Int. Conf. on Machine Learning (ICML)*, pages 9269–9278, 2020.

[43] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. In *Proc. 34th Int. Conf. on Machine Learning (ICML)*, 2017.

[44] M. Taufiq, P. Blöbaum, and L. Minorics. Manifold Restricted Interventional Shapley Values. In *Proc. 26th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[45] K. Vafa, Y. Deng, D. Blei, and A. Rush. Rationales for Sequential Predictions. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[46] G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu. On the Tractability of SHAP Explanations. *Journal of Artificial Intelligence Research (JAIR)*, 74:851–886, 2022.

[47] S. Wäldchen, J. Macdonald, S. Hauch, and G. Kutyniok. The Computational Complexity of Uderstanding Binary Classifier Decisions. *Journal of Artificial Intelligence Research (JAIR)*, 70:351–387, 2021.

[48] X. Xuan, P. Xizhou, L. Nan, H. Xing, M. Lin, Z. Xiaoguang, and D. Ning. GAN-Based Anomaly Detection: A Review. *Neurocomputing*, 493, 2022.

[49] J. Yi, E. Kim, S. Kim, and S. Yoon. Information-Theoretic Visual Explanation for Black-Box Classifiers, 2020. Technical Report. https://arxiv.org/abs/2009.11150.

[50] J. Yu, A. Ignatiev, P. Stuckey, N. Narodytska, and J. Marques-Silva. Eliminating The Impossible, Whatever Remains Must Be True, 2022. Technical Report. https://arxiv.org/abs/2206.09551.

[51] O. Zaidan, J. Eisner, and C. Piatko. Using "Annotator Rationales" to Improve Machine Learning for Text Categorization. In *Proc. Conf. North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 260–267, 2007.

[52] W. Zhao, S. Alwidian, and Q. Mahmoud. Adversarial Training Methods for Deep Learning: A Systematic Review. *Algorithms*, 15(8):283, 2022.

[53] L. Zintgraf, T. Cohen, T. Adel, and M. Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In *Proc. 7th Int. Conf. on Learning Representations (ICLR)*, 2017.